Index

Entries in **boldface** are menu commands or dialog boxes. Entries in `typewriter face` are functions, methods, variables, keywords, or files.

Numbers

4-byte IEEE single precision 306, 590
8-byte IEEE double precision 306, 590
10-byte SANE extended precision 306-307, 336, 590
12-byte MC68881 extended precision 306-307, 590
12-byte universal extended precision 306-307, 590

A

additive operators 593
address 395
  absolute 257-258, 372
  of resource 659
  register 265, 266, 316, 597
  relative 257-258, 285
aggregate initializer 268
aliases 32, 156, 159, 182
ANSI standard 5, 297
  conformance to 329-330, 461, 587-606
    additive operators 593
    bitwise shift operators 593
    cast operators 593
    character constants 591
    character sets 588
    conditional inclusion 595
    declarators 595
    diagnostics 588
    enumeration specifiers 595
    expressions 592
    floating and integral 592
    floating types 592
    header names 592
    identifiers 589
    linkages of identifiers 589
    macro replacement 596
    multibyte characters 588
    multiplicative operators 593
    `#pragma` directives 596
    program execution 588
    program startup 588
    signed and unsigned integers 592
    `sizeof` operator 593
    sizes of integral types <limits.h> 588
    source file inclusion 596
    storage-class specifiers 594
    structure and union members 592
    structure and union specifiers 594
    `switch` statement 595
    translation phases 588
    type qualifiers 595
    types 589
  implementation-defined behavior identified by 587
  undefined behavior identified by 587
Apple
  Extended Keyboard 171-172
  Installer 21
AppleEvents 261, 719
AppleScript 30-31, 143
`AppleTalk.h` 354
application 137-138, 255
  See also program, project

building 256-262
segments in 139
archives 21, 23
arguments
char-sized 373
converting declarations from Pascal to C 352
passing to Toolbox routines 352
pushing 372, 374, 376
strings as 353
array 457
alignment of 321
assigning to 723
bounds 90, 705, 739
`char` 370
declaring labels within 552-553
dimensionless 315, 605
examining and modifying 86-94
initializing in a type statement 720
nesting 720
of `int`s 265
value of 86
window 90, 242-243
arrow keys 171
assembly language 379-398
and the Toolbox 389-392
declaring local storage 396
instructions per line 381
instructions referring to variables 396
mixing statements with C code 381
multi-line macros 397-398
using C identifiers in 384-386
using constants 396
using instructions with labels 386-388
writing code 381-384
auto-make facility 137, 206-207

B

Balance command
behavior
implementation-defined 587
undefined 587
bit-field 594, 744
bits, setting 260-262
bitwise shift operators 593
bomb alert 697, 703-704, 706
Boolean values 362, 392, 537, 539, 558
braces 169, 312
brackets 169
breakpoint
clearing 79, 86
saving 246
setting 5, 77-79, 87, 102, 228-233
using statement markers to set 74
bug alert 201-202, 697

C

C code
See also code
compiling 327-340
declarations vs. Pascal declarations 352
labels 386-388
libraries. *See* libraries
mixing with C++ code 303
routines 371-375
writing 297-326
C++
applications, building 691
code
mixing with C code 303
writing 5
call chain 74, 223
calling conventions
C 372
Pascal 351, 362, 372, 375-377
calls 272-273
cast operators 593
character
escape 567, 580
literal 729
multibyte 588
newline 567
strings 592
class(es)
See also object-oriented programming
abstract 105
browser 173, 287-294
keyboard shortcuts for 293
constructor 173
declarations 104-107, 289-290
definitions of 173
examining 86-94
families of 105
hierarchy 105, 289
indirect 322
libraries. *See* libraries
modifying 86-94
object pointer of 108
objects of 105-106

`Close` call 272
code
See also C code, C++ code, object code, source code
ANSI-conformant 312, 329-331
C++-compatible 330-331
compiling 144, 327-340
debugging. *See* debugger
optimized 246-247, 324-326, 341-348
overflow 729
owning resource 284
porting 367-378
processor-specific 335-336
referencing 255
resources 137, 139, 255
building 274-284
calling 280
headers for 280-283
libraries, using in 278-279
locking 279-280
`main()` routine, writing for 276-277
merging into files 283
reasons to write 274
segments in 276, 278, 283
variables in 277-278, 362
segments 729
stepping through 5, 74, 77, 100, 119, 221, 234
commands
Add (**Source**) 120, 135
Add Files (**Source**) 45, 57, 120, 136
Add Window (**Source**) 663
Arrange (**Windows**) 175, 671
Attach Condition (Debugger **Source**) 231-232, 682
Balance (**Edit**) 169
Bring Up To Date (**Project**) 403, 660
Browser (**Source**) 289, 670
Build (**Project**) 255-256
Build Application (**Project**) 48, 65, 120, 138, 661
Build Code Resource (**Project**) 661
Build Desk Accessory (**Project**) 138, 661
Build Device Driver (**Project**) 661
Build Library (**Project**) 127, 255, 285, 661
Categories (**Reference**) 409
Check In (**SourceServer**) 474, 477
Check Link (**Project**) 205, 660-661
Check Out (**SourceServer**) 475, 480
Check Syntax (**Source**) 206, 667
Clear (Debugger **Edit**) 678
Clear (**Edit**) 81, 86, 90, 221, 238, 617
Clear All Breakpoints (Debugger **Source**) 86, 229, 246, 682
Clear All Expressions (Debugger **Data**) 238, 246, 683
Clear Breakpoint (Debugger **Source**) 681
Close (Debugger **File**) 677
Close (**File**) 176, 583, 614
Close All (Debugger **File**) 243, 677
Close All (**Windows**) 177, 672
Close & Compact (**Project**) 142, 655
Close & Compact Project (**Edit**) 620
Close Project (**Project**) 66, 134, 135, 655
Compile (**Source**) 42-43, 46, 59, 61, 120, 205, 520, 667
Context Free (Debugger **Data**) 240, 685
Copy (Debugger **Edit**) 678
Copy (**Edit**) 617
Copy Code Examples (**Edit**) 408
Copy to Data (Debugger **Edit**) 237, 678
Cut (Debugger **Edit**) 678
Cut (**Edit**) 616
Data (Debugger **Windows**) 685
Debug (**Source**) 81, 224, 227, 665
Disassemble (**Source**) 210, 378, 667
Edit 'filename' (Debugger **Source**) 682
Edit Windows (**Windows**) 672
Enter Selection (**Search**) 650
ExitToShell (Debugger **Debug**) 93, 243, 250, 681

filename.c (Debugger **Windows**) 685
Find (**Search**) 183, 189-190, 648
Find Again (**Search**) 183-186, 189-190, 650
Find in Next File (**Search**) 189, 651
Find in THINK Reference (**Search**) 174, 405, 652
Full Titles (**Windows**) 672
Func Ref (**Reference**)
Get Info (**Source**) 665
Go (Debugger **Debug**) 234, 679
Go To Line (**Search**) 172-173, 652-653
Go To Next Error (**Search**) 204, 654
Go To Next Match (**Search**) 188
Go To Previous Error (**Search**) 204, 654
Go To Previous Match (**Search**) 189
Go Until Here (**Debugger Debug**) 235, 680
Header Files (**Reference**) 410
Help (**Reference**) 409
Inspector (Debugger **Debug**) 681
Launch SourceServer (**SourceServer**) 473
Load Library (**Source**) 59, 137, 667
Locked (Debugger **Data**) 91, 240, 685
Make (**Source**) 208, 357, 668
Mark (**Search**) 179, 653
Modify Read-Only (**File**) 477, 615
Monitor (Debugger **Debug**) 249, 681
New (**File**) 41, 120, 582
New Project (**File**) 129
New ProjectorDB (**SourceServer**) 473
Open (**File**) 104-105, 120, 165-166, 582, 613
Open Project (**File**) 71, 131, 133-134, 285
Open Selection (**File**) 167, 614
Options (**Edit**) 144, 617
Page Setup (**File**) 583, 615
Paste (Debugger **Edit**) 678
Paste (**Edit**) 617
Precompile (**Source**) 214, 357, 667
Preprocess (**Source**) 209, 667
Print (**File**) 174, 583, 615
Printing 615
Project Window (**Windows**) 672
projectname (Debugger **Windows**) 685
Quit (**File**) 48-49, 66, 615
Remove (**Project**) 157
Remove (**Source**) 55, 137, 664
Remove Marker (**Search**) 654
Remove Objects (**Project**) 128, 133, 142-143, 660
Replace (**Search**) 650
Replace All (**Search**) 651
Replace & Find Again (**Search**) 651
Revert (**File**) 615
Run (**Project**) 46, 59, 62, 72, 80, 97, 100, 120, 199, 662
Save (**File**) 120, 176, 582
Save a Copy As (**File**) 177, 615
Save All (Debugger **File**) 246, 248, 677
Save All (**Windows**) 177, 672
Save As (**File**) 42, 61, 156-157, 582
Select All (**Edit**) 617
Send Command (**SourceServer**) 495
Set Breakpoint (Debugger **Source**) 229, 682
Set Context (Debugger **Data**) 91, 240, 683
Set Project Type (**Project**) 74, 137, 251, 257, 263, 276, 285, 655
Set Tabs & Font (**Edit**) 170
Shift Left (**Edit**) 169
Shift Right (**Edit**) 169
Shortcuts (**Apple**) 677
Show Condition (Debugger **Source**) 233, 682
Show Context (Debugger **Data**) 91, 240, 683
Skip To Here (Debugger **Debug**) 235-236, 247, 680
SourceServer (**Source**) 666
Step (Debugger **Debug**) 234, 679
Step In (Debugger **Debug**) 78, 234, 679

Step Out (Debugger **Debug**) 78, 234-235, 679
Stop (Debugger **Debug**) 235, 680
Switch to Project (**Project**) 134, 655
Trace (Debugger **Debug**) 76, 234, 680
Undo (Debugger **Edit**) 678
Undo (**Edit**) 168, 616
Use Debugger (**Project**) 71, 100, 125, 217, 248, 662
Zoom (**Windows**) 672
comments 311, 313
C++ 603
Communications Toolbox 351
compiling 206-207. *See also* source files, C code, code, THINK Project Manager
concatenation 566
constants 721, 730-731
character 591-592
enumeration 734
floating-point 736
integer 591-592
literal 561
numeric 561
constructor 173, 448-450, 463, 718, 734, 748
`Control` call 272-273
creator 64, 138, 257, 263
CSE elimination (common subexpression elimination) 246
current
function indicator 223
statement arrow 74, 78-79, 103, 220, 222-223, 679

D

Data menu 683-685
Data window 72-73, 80-91, 220-221, 236-243, 683
databases, finding lost 410-411
Debug menu 73-74, 679-681
debugger, source-level 5, 69-94, 119, 215-252
cache space used by 227
clearing values of expressions 84
commands used by 233-236
display formats for expressions 84, 91-93
error messages in 243-245
evaluating expressions 91
executing functions 76
memory required for 215, 710
menus 675-686
optimized code, using with 246-247
quitting 93
solving problems with 709-710
state 246
turning on 71, 100, 117
using escape characters 92
windows 72, 80-81, 101
debugging, 59-62, 143, 324, 336-338
See also debugger
examining variables during 80-86
tables for source files 72
declarations
parameter 362
Pascal 352
type 536
used for writing Macintosh programs 355, 362
desk accessories 138-139, 255
building 262-274
calling 270, 272
default header for 270
locking 270
number of segments in 139
variables in 362
destructor 448, 450-451, 463, 718, 748
device drivers. *See* drivers
Device Manager 264, 268, 272
dialog
Add 45, 57
Add Files 136, 664
Application 656
Arrange 671
Build 255-256, 661
Build Code Resource 283
Check In 478-480
Check Out 480-486
Desk Accessory 658
Device Driver 658
Find 183, 649
Go To Line 172-173, 652-653
#INCLUDE Paths 577
Make 207-208
Mark 179, 653
Multi-File Search 187
Name Revisions 486-487
New Project 129, 692

New ProjectorDB 473
.o Converter 524, 628
Progress 200
Remove Marker 181, 654
SADeRez 574
Segment attribute 141
Set Project Type 64, 256, 263, 274, 276, 283, 318
Set Tabs & Font 170
Symantec C++ 223
Symantec C++ Easy Install 28
THINK C options 223, 300, 302, 308, 310-312, 329, 339, 344, 347, 353, 369, 459, 628
THINK Project Manager options 144-145, 241, 247-249, 618-630, 649
THINK Rez Options 568-569, 645
Types Files 575
Update 47
directives
`DC.B` 394
`DC.D` 395
`DC.S` 395
`DC.W` 394
`DC.X` 395
`#define` 558
`#elif` 595, 599, 738
`#else` 738
`#endif` 604, 738
`#error` 588
`#if` 595
`#ifdef` 738
`#pragma` 290-300, 315-317, 332, 357, 455, 596-599
`#undef` 558, 599
drivers, device 137
building 262-274
calling 272
definition of 262
glue 264, 266-267
header, setting fields of 268-270
how they work 264-265
libraries, using in 268
`main()` function of 264-267, 274
arguments of 265, 271
returning from 271-273
segments in 139, 264-265
variables in 266-268, 362
dynamic
binding 108
storage 237

E

Edit menu 616-647, 678
Edit window 41, 120, 125, 165, 174, 667-668
encapsulation 418. *See also* object-oriented programming
end key 171-172
enumeration types 312, 320, 734
error messages 43, 59-60, 202-205, 243-245
See also Appendix C
categorization of 714
escape
characters 567, 580, 749
sequence 567, 735
event 260-263
activate 77, 88
update 77
Event Manager 277
executable
code 224
file 117, 148
expressions 562-563, 592
calculation of size 723
changing context of 91, 240
compiling in a context 90-91
default formats of 683-684
entering 90-91
integer constant 747
locking 91, 240-241
lvalue 755
saving 246
too complex 736
values of, formatting 238-239
values of, setting 238
with global scope 82, 91, 237
with local scope 82, 91, 237
`extended` type 365
extensions 310-314, 319-320, 631, 726
adding 147
changing translator for 147
deleting 148

F

file
See also source files, project, resource files, THINK Project Manager
building code resources into 283

closing and saving 176-177
controlling changes in during team development 469-504
creating and opening 165-168
definitions, looking up 173-174
duplicate names of in trees 156
editing of 163-196
examining limited area of 172
lines, numbering of 172
managing 158-160
marker in 653, 722
moving
on disk 156-157, 207
to another machine 157
moving around in 170-173
object 517-525
organizing 155-156
output description 578-579
resource 117
resource description 529-570
Save As 176-177
shared by projects 207
text 165-166
type, setting 257
vocabulary 519, 521
File menu 611-615, 678
Finder 5, 420-421
flag 268
partition 259
SIZE 259
floating-point representation 305-310, 319, 336, 364-365, 386, 404, 590, 591, 637-638, 657, 740. *See also* option, Native floating-point format
folder
adding 577
`Aliases` 158-159
`Apple #includes` 358-359
`AppleScript Utilities` 31
`Demos` 53, 69
`Development` 24, 38, 53, 69
files, placing in 160
files with same names in 156
`Mac #includes` 356-357
folders and files in 358
naming 133
nesting 155-156
`(Old Files)` 161
`Online Documentation` 32
`oops` Libraries 426
opening 136
Preferences 695
project-specific 158
`Projects` 39
referencing 299
removing 577-578
shielded 158
`Standard Libraries` 43-44, 279
System 31
`THINK #includes` 359-360
tools 32
top-level 155
font settings, changing 170
Frontier scripts 143
function
See also routine
abort 602
ANSI library 707-708
arguments of 372
arguments to 720, 754
as members of a class 435, 438. *See also* object-oriented programming
assigning to 723
built-in 551-552
calling 316
from the Data window 241
class method 439
clock 603
crashing when calling 705
`Debugger()` 233
`DebugStr()` 233
declaring with argument list 719
definition of 173
entry points into 389
exit 602
fflush 601
fgetpos 602
fmod 600
fprintf 602
fscanf 602
ftell 602
getenv 602
`GetHandleSize()` 241
`GetNextEvent()` 235, 680
getting addresses 374
in profile libraries 514-515
inline 225, 311, 313, 357, 360, 596, 604
keys 171-172
Macintosh Toolbox 708-709. *See also* Toolbox

main() 264-267, 659
math 709
memory management 602
nonvirtual member 438
perror 602
popping arguments off stack 347
prototypes 311, 314, 322-323, 333-335, 354, 360, 705-707, 720
reference page 407-408
remove 601
rename 601
returning struct, union, or double 373-374
returning the unexpected 706-709
signal 601
stepping into 75
stepping out of 75
strerror 603
string-handling 403
string values of 563
system 603
tracing of 510-511
using your own 707
utility 364
virtual method 438
WaitNextEvent() 235
writing to behave as if written in Pascal 362

G

global optimizer 324-325, 344-346
glue 351
Grep 118, 182, 650
error 737
examples of 194-196
searching for 190-196

H

handle 433. *See also* object-oriented programming
header file 117, 299, 357-364
See also precompiled headers
AppleTalk.h 354
contents of 209
editing 5
for device driver 268-269
including in more than one source file 226
names
list of 167
preprocessing 592
once-only 299-300
oops.h 259
opening 166-167
placement in folder 160-161
processing 559
SetUpA4.h 267, 278-279, 281, 362
viewing in the source window 224-226
hexadecimal constant 393
home key 171-172
HyperCard XCMDs 351
hyperlinks 406, 410

I

identifiers 299, 558
See also relevant error messages in Appendix C
if-then-else processing 560-561
implementation-defined behavior 587-606
#include files 44, 59, 212-213, 557, 746. *See also* precompiled header files
#include statement 214, 299-300, 355, 357, 407, 559, 570, 576, 644, 715
indentation, editing 169
inheritance 419-420. *See also* object-oriented programming
INIT conflicts 699-701, 710
inline
assembler 279, 281, 311, 313, 359, 360, 379-398, 509, 604
calling traps 392
case sensitivity of 389
errors generated by. *See* Appendix C
vs. other assemblers 393-395
function 311, 313-314, 357, 360, 596-597
integers
representation of 302-305, 319, 539, 589
signed and unsigned 592
interface, AppleTalk 354
internal representation 343
I/O
parameter block 265
queue 272-273

J

`jIODone` problem, in returning from drivers 272-273
jump table 148-152, 318, 656-657, 754
  contents of 257
  entry 152, 372
  size of 257

K

keys 171
  arrow 171, 293
keywords 311, 313, 428, 443-444

L

label 551-557
  assembler 386-388
  declared as extern 389
  declared within arrays 552-553
  defining 553
length modifier 541
libraries
  adding to projects 44-46, 118, 120, 127
  ANSI 44, 118, 127, 174, 365, 374, 401, 507, 612, 703, 707
    corrupted 707
  ANSI++ 401
  ANSI-A4++ 402-403
  ANSI-small++ 402
  AppleTalk 351
  binary 127
  building 284-285
  CommToolbox 351
  Complex 5, 118, 174, 401
  CPlusLib 401
  CPlusLib-A4 402
  device drivers, using in 268
  HyperXLib 351
  IOStreams 5, 118, 174, 401
  loading 46, 97, 137
  MacTraps 59, 71, 118, 268, 279, 351, 352, 703
  MacTraps2 118, 268, 279, 351, 352
  moving 157
  MPW .o 127
  nAppleTalk 351
  object size of 46
  oops, oopsA4, oopsDebug, oopsFar 426
  OSL 351
  profile 514-515
  profile++ 514-515
  project 127
  QuickTime 351
  SANE 351, 364, 365
    values used by 365
  `standard` 43-44, 279
  standard with C 399-412
  THINK tree, keeping in 155
  UNIX 118, 401
  UNIX++ 401
link
  errors 44, 174, 201, 205, 403, 427, 588, 754
  map 209-212, 621
literals 561, 562, 729, 750
loop 99
  event 77
  `for` 234

M

`MacHeaders` 213-214, 265, 333, 355-357, 358, 390, 644, 703
  editing 356-357
  files contained in 355, 360
  files not contained in 355-356
`MacHeaders++` 213-214, 265
Macintosh Programmer's Workshop (MPW) 127, 177-178, 517, 569, 571
Macintosh routines treated as if Toolbox traps 390
Macintosh Toolbox. *See* Toolbox
macros 209, 281, 303-305, 307-310, 533, 569, 644
MacsBug 249, 324, 337, 643
MacTraps. *See* libraries
mangled names 377-378
markers 179-181
`main.c` 55
  deleting 55-56
`main()` function 73
math functions, trouble with 709
memory, insufficient 702-703
menus
  Data 683-685
  Debug 73-74, 679-681
  Edit 616-647, 678
  File 611-615, 677
  Project 655-662, 681-682
  Search 32, 648-654

Source 663-670, 682
Windows 671-672, 685
messages. *See* error messages and warning messages
methods
calling 100
definitions, looking up 173, 290-292, 421
implementing 105-106
overriding 105, 419
relationship to messages 110
virtual 105, 257, 322
missing resource error 703
MPW .o 127
MultiFinder 5, 39, 49, 63, 69, 100, 201, 260-261
multiplicative operators 593

N

nonprinting characters 186
note files, adding to projects 129
null byte 658
nVIR virus 699-700

O

.o Converter 127, 517-525
conversion limits of 525
setting options for 523-524
object
class determined when 108
code
corrupted 143, 702
generation 123
linking 66, 117, 119
making as small as possible 119
placement of 206
size of 123, 125
translating from source files 127
components, of source files 148-152
CODE, DATA, JUMP, STRS 148
creating 107-108
debugging 5
declaring as a pointer 745
extensions 423-466
files 517-528
passing 352
pointer to 352
response to message 104-107
object-oriented programming 310, 415-422
abstract class in 421
allocator 451-454
basics of 95-114
class.
See also types of class in this entry
accessing members of 433-435
allocator 433
declaration of 104-107, 426, 428-430, 433, 729
declaring members of 435-441
defining new 108-110, 418, 435, 455, 725
direct 458
indirect 458, 459-460
instances of 421
member functions of 427-428, 438
organizing 455
private member 429, 444-446
protected member of 444-448, 716-718
public member 429, 444-446
variable 437
constructor 173, 448-450, 463, 718, 734, 748
deallocator 433, 451-454
destructor 450-451, 463, 718, 748
document 612
dynamic binding 108
encapsulation 418
handle 433
inheritance 108
instance variables 435-436
message
definition of 417
responding to 419
sending 100, 420-421
object
allocated as handles 427
allocated as pointers 429
and the Macintosh interface 420-422, 455-458
assignment, rules for 432-433
creating 107-108
current 428
declaring and using 430-433
definition of 417
deleting 430-431
locking down 433
reference 427, 430-433, 463
storage 752
value 463

working with 421-422
root classes in 419, 421, 429, 459
subclasses in 419, 421, 429, 432, 436
superclasses in 108, 419, 432
tips and techniques for 454-458
vs. procedural programming 417, 420, 422
Object Pascal 425
once-only headers. *See* header file
online *Standard Libraries Reference* 404-411. *See also* THINK Reference
`Open` call 272
operator preference 562-563
operators 593
unary 747
optimizer 341-348
definition of 343
global 343, 344-346
internal representation, editing of 343
local 343, 347-348
when not to use 343-344
option
Align arrays of char 638
Align even-length char arrays within structs 598
Allow dataview function calls 237, 241, 248, 625
Always check file dates 182, 207, 621
Always compact projects 142, 620
Always generate stack frames 223, 337, 382, 509, 598, 643
Always resegment on overflow 621
Always save session 248, 625
Always segment on overflow 139
Assign variables to registers 598
Attrs 660
Automatic register assignment 300, 302, 382, 594, 640
Balance 647
Batch search 184, 188, 623
Check pointer types 329, 330, 332-333, 598, 630
Classes are indirect by default 433, 598, 634
Code motion 346, 641
Confirm project updates 620
Confirm saves 623, 659
CSE Elimination 246, 345, 641
Custom header 280
Dead variable elimination 247
Decode trigraphs 598
Defer & combine stack adjusts 247, 347, 382, 598, 639
`#define` _STDC_ 329, 330, 629
Don't ask for project on open 135, 621
8-byte doubles 305, 308, 369, 636
Entire word 184
`enum`s are always `int`s 312, 329, 330, 595, 630
Exclude <System> files 188, 623
Far CODE 257, 258, 372, 656
Far DATA 258, 656
4-byte ints 304, 331, 369, 371, 595, 635
Function prototypes are required 598
Generate class names 635
Generate code for 68020 598
Generate code for 68881 598
Generate Link Map 210, 621
Generate MacsBug names 378, 598
Generate MC68020 instructions 244, 323, 383, 633
Generate MC68881 instructions 244, 301, 306, 307, 308, 323, 336, 365, 374, 383, 404, 634
Generate profiler calls 338, 507, 598
Grep 184
Honor register declarations 598
ID 659
Induction variable elimination 345, 641
Infer function prototypes 598
Language extensions 310, 329, 330
MacsBug names 338
Methods are virtual by default 598, 635
Multi-file search 184
Multi-segment 658, 659
Name 658, 659
Native floating-point format 306, 307, 308, 331, 336, 365, 369, 371, 374, 637-638
Optimize monomorphic methods 440, 621

"\p" is unsigned char [] 369, 598, 636-637
Partition 656
Projector-aware 623
Prefix string 569, 645
Recognize trigraphs 312, 329, 330, 629
Redeclared types ok 568, 645
Register coloring 346, 641
Reopen files 166, 623
Resource alignment 568, 645
Run global optimizer 599
Select appropriate size for enum types 598
Separate STRS 148, 259, 657
Set tabs & font 646
Shift left 647
Shift right 647
Show CODE size 149, 627
Show DATA size 149, 627
Show JUMP size 149, 627
Show segment numbers 149, 627
Show STRS size 149, 627
SIZE flags 656
Smart link 256
Store debug info separately 142, 249
Strict prototype enforcement 213, 323, 329, 330, 333-335, 631
Suppress redundant loads 348, 598, 640
THINK C + objects 458
Type 658, 659
Update program windows 248
Use Debugger 217
Use external editor 182, 623-624
Use function calls for inlines 509
Use global optimizer 344, 640
Use long format for MacsBug names 598
Use second screen 218, 248
Use source debugger 248, 624
Use toolbox trap list 519, 524, 628
Use '.v' File 519, 524, 628
Whole words only 622
Wrap 622
Wrap around 184
output description file 578

P

page up/down keys 171-172
panic button 79
parameters 362-363
parentheses 169, 191
partitions 5, 63
See also program, project, THINK C Project Manager
size of 64, 260
Pascal
calling conventions 351, 362, 363, 372, 375-377
calling routines of indirectly 363
function entry 376-377
function exit 377
patterns. *See* Grep
pointer 265, 278
arithmetic operators used with 742-743
array of 90
event record 265-266
stray 201
to array 741
to function 741
to objects 352, 427
to random memory 201
to value of class type 430
types of 332-333, 761
unitialized 143, 705
polymorphism 419-420. *See also* object-oriented programming
`#pragma` directives 299-300, 315-317, 332, 357, 455, 596-599
precompiled header files 212-214, 332-333, 355-357, 644, 715, 732, 759
benefits of 212, 710
creating your own, 213-214
material contained in 212
preprocessor
directives 317-326, 390, 557-561, 743
output 209
`Prime` call 273
profiler. *See* program
program
See also project
building 35-50, 64, 120
counter 249
editing with the optimizer 343-348
executing line by line 74
linking 49, 65, 200
making as small as possible 65
partition size of 705, 710
profiling 505-515

tracing functions 510-511
running 79, 118, 197-214, 218
sticking in loop 79
stopping 79
writing with THINK Project Manager 115-120
project
See also resource(s), file
adding files to 56-59, 123, 129, 135, 519-521, 613
back-up, using 143
building from scratch 37-50 (tutorial), 53, 119, 129-133, 253-286, 691-692
closing 66, 135
compiling files in 59, 702, 703-704
components. *See* Object components
compressing 142
corruption of file 702
crashing while running 704-706
Creator 64, 138, 257, 263
definition of 117, 121, 123
document 59, 72, 118, 129
compacting 620
files, types of in 126
folders, placement of
libraries, adding to 44-46
library, used as 127, 284
model 129-131, 611, 612
creating your own 133
naming 132
opening 37, 63, 133-135, 613, 701-703
options for 143-145
partitions 63, 119
size of 64
prefix 204, 338-340, 355
recompiling 702. *See also* project compiling
recovering corrupted 142
removing files from 55-56, 137
resource file 128-129, 529-570
adding resources to 531
syntax of statements in 534-550
running 46-48, 59, 120
segmenting, 124, 139-141, 726
selecting files in 137
sharing files between 158
size of 151
stationery pad 129, 131-133
stopping at breakpoints 87
turning into application 48-49
type of
effect of changing 138, 268
setting 137-138, 262, 275
updating 47, 59, 62, 118, 120, 199
using built-in project models 38
window 39, 43, 46, 72, 97, 100, 117, 123-126, 130, 148, 165, 217
bringing to front 81
information shown in 148
Project menu 655-662, 681-682
Projector database. *See* SourceServer
prototypes 333-335, 597, 631-632
C 357
for `char`-sized argument 373

Q

`QuickDraw.h` 355, 359

R

references
A5-relative 724-725
circular 440
computed 725
forward 440
16-bit data 724
32-bit code 724
registers 300-302, 348, 385, 393, 594, 716, 734, 746, 751
relational operators 593
representation
floating-point 305-310, 319, 336, 364-365, 386, 404, 590, 591, 592, 637-638, 657, 740
See also option, native floating-point format
of integers 302-305, 319
text 573
ResEdit utility 66, 128, 529, 704
Resorcerer 128, 529
resource(s)
`'ALRT'` 532
alignment 645
`'CODE'` 138
compiler 529-570
`'DCOD'` 274
decompiler 532
declarations 532
definitions of in output file 578-579
description file 529-570, 573, 575

'DITL' 532
'DRVR' 138, 274, 532, 658
file
code size in projects 128
comments in 533
copying into an application 66
corrupted 704
description 529-575
editor used to create 128
preprocessor directives in 533
statements in 532-550
structure of 532-534
translating from source file 127
used for 128
using 66
'FOND' 532
'FONT' 532
'FWID' 532
'INTL' 532
'MENU' 532
'NFMT' 532
owning 274
'PICT' 532
QuickDraw 551
'SIZE' 259
statement 547
types 532-570
choosing for decomposition 580-581
'WIND' 546
Resource Copier 126, 531
Revert command 168
routine
See also function
address of 257
AppleTalk
calling 354-355
assembly language 381
C 371-375
C++ 377-378
callback 267, 276, 362, 375
calling another routine 257
Close 271
event-fetching 79
File Manager 392
for converting strings 354
for working with globals 359
glue for 264, 266, 351
in Binary-Decimal Conversion Package 360
library 44, 241
main() 73, 220, 271, 276, 277, 280
Memory Manager 457
nonstatic 372
Open 266, 271
operating system 6
Pascal 362, 375-377
QuickDraw 353
run-time 525
SANE 365. *See also* SANE, function
TextEdit 390
Toolbox 6, 16, 241, 349-365, 433, 612
C prototypes for 357
calling 351-357
passing arguments to 352-353
using in assembly language 389-392
trap intercept 267
utility 401
run-time error 243, 587

S

SADeRez 533, 571-584
running 573-574
saving and restoring options for 581-582
setting options for 579-581
SANE (Standard Apple Numerics Environment) 361, 364-365
Script Editor 31-32
Scriptable Text Editor 31-32
scripting 695
search, setting characteristics of 649-650
Search menu 32, 648-654
segment of project 124, 139-141, 264, 692
collapsed 124-125, 137, 140
expanded 124-125, 140
moving files into 140
number of 139
rearranging 141
size of 139
viewing numbers 150
shielded folder 158
sizeof operator 593
smart linking 255, 284
source
code 118
coordinating with SourceServer 467

looking at 209
translators for 127
files
adding text to 644
adding to projects 56-59, 120
assembly code produced by 209
changing 48
compiling 5, 42-44, 48, 59, 97, 117-118, 206-212
creating 41-42, 117-118, 120
debugging 72. *See also* debugger, source-level
dependencies between 123
editing 5, 118, 123, 226
errors in 203
link map 123
moving 156-157
object components of 148-149
opening 166
placement in folders 161
preprocessor output of 209
projects, number in 123
saving 120
size of 39, 59
suffixes of 206
syntax, checking 43
translating into object code or resources 127
text 221
window
changing files in 79-80
contents of 219-220
context of expressions in 90
selecting statement in 222-223
status panel in 73, 220
viewing header files in 224-226
Source menu 663-670, 682
SourceServer 467-504
command line interface 494-504
database server used by 467, 469, 487-489, 492-493
file handling by
checking in 474-475
checking out as read-only 476-477
checking out for modification 470, 474-476
revision tree of 469-471, 481, 486-487, 489-491
key capabilities of 471-472
limitations of 472
project handling by 473-474
tutorial for 473-477
stack
collision 704
during a C function 373
during a Pascal function 376-377
errors 706
frames 223, 324, 337, 643, 722
statement 735
See also type
array 723
assignment 237
`break` 721
`byte` 538
C branching 387
`continue` 731
current 222, 237
executing 680
`data` 535, 537, 548, 732
data-type 537
declaration 732
`#include` 214
markers 74
`rect` 743
resource description 532-550
sample type 546
selected 222
stepping through 74
switch 595, 723, 735
tracing 76-77
stationery pad 131-133
`Status` call 273
stepping through code 5, 74, 77, 100, 119, 221, 234
string 182-190, 538, 694, 763
accepted as operands by directives 394
address 684
C 541, 684
character 592, 684
converting one form to another 354
defining as C 394
floating-point 684
hexadecimal 684, 737
looking for in more than one file 186
null-terminated C 353
Pascal 353-354, 540, 566, 591, 636, 684
patterns of 190-196
pointer 684
prefix 645

replacing 185
signed decimal 684
terminating 769
truncating 540
types of 565-566
unsigned decimal 684
value 563
word 541
string literal 148, 259, 278, 321, 353, 370, 591, 636, 637, 657
struct
`char` arrays in 370
examining and modifying 86-94
fields of 88
value of 86
window 89-90, 242-243
Symantec C++
See also Project and THINK Project Manager
calling functions from THINK C code 377-378
code resources created in 275
compiling 5
compression into archives 21
databases 32
debugging. *See* debugging
installing 6, 19-32
insufficient space for 24-26
virus protection programs 22
libraries available with 174
options dialog 144
porting code to 370-371
system requirements for 6
translators for source code 127, 144
user interface generator 5
symbolic names 550
syntax 43, 382, 534-550, 561-568, 764
System 6 5, 39, 49, 63, 69, 100, 118, 201, 264
System 7 5, 32, 39, 49, 63, 69, 100, 118, 134, 156, 181, 201, 260-262

T

tab
characters 186
settings, changing 170
THINK C
absolute address extension of 360
access control mechanism in 443, 445
assembly language, using with 379-398
calling conventions understood by 371-372
code, writing 297-326
compiling by 205, 318-319, 335-338
compression into archives 21
databases 32
debugging with 336-338
enhancements made to environment 689-695
errors generated by. *See* Appendix C
extensions to C defined by 310-314, 319-320, 631, 726
floating-point numbers 305-310
header files
modifying for 358-359
rules of 299-300
identifiers in 299
implementation-defined behavior 587
implementation of C 299-302
installing 19-32
object-oriented extensions to 423-466
optimizer 341-348
porting code to 369-370
registers, loading data into 348
standard libraries of 399-412
system requirements for 6
type checking, enforcement of 331-335
upgrading 691
variables, allocation to registers 300-302
virus protection programs and 22
vs. THINK C++ 461-464
THINK editor 181-196
THINK environment 5, 117
THINK Project Manager
adding files with 45, 57
building applications in 35-50
building libraries with 284-285
compatible editors, using with 181-182, 207
configuration management. *See* SourceServer

debugging with. *See* debugger, source-level
definitions, looking up 173-174
editing file extensions 146
errors generated by. *See* Appendix C
file extensions recognized by 126, 136, 146-147
files, keeping track of 59, 128
folders, looking for files in 155
launching 37, 701-703
link maps 210-212
location of 156
note files 129
online *Standard Libraries Reference* 404-411
options dialog 144-145
partition size of, changing 702
quitting 66
re-installing 702
returning from drivers 272
running programs from 197-214
search facilities in 118
segmenting projects by 692-693
selecting files with 137
smart linking, used by 255-256
text editor 42, 51-68, 123
translators used by 126-127, 146-148
tree 39, 155-158
troubleshooting with 697-711
updating projects with 47, 199
vs. other development environments 117, 123
writing programs with 117-120
THINK Reference 16
configuring databases 31-32
hyperlinks in 406
pages in 409-410
switch to 174
THINK Rez 66, 126, 205, 217, 529-570
errors generated by. *See* Appendix C
options for 568-569
vs. MPW Rez 569-570
THINK tree 39, 155-158
tiling 176
TMON 643
TMON Pro 249
Toolbox 13, 16, 107, 118, 173, 234, 266, 279, 347, 349-366, 612, 708-709
official interfaces to 358
using in assembly language 389-393
ToolServer 177, 179
tracing 234
translators 126, 147
trap 351, 357
modifier, 391-392
bits 360
names 389-390
tree
project 155
deleting files from 157
keeping source files in 155
putting folder aliases in 158
rules regarding use of 155-156
THINK Project Manager 39, 155
putting folder aliases in 158
rules regarding use of 155-156
trigraphs 312
tutorials
Bullseye 69-94
Hello World++ 35-50
LearnOOP 95-114
MiniEdit 51-68
two's complement binary numbers 302, 589
type
align 542
array 543
`boolean` 539
character 540, 728
checking 331-335
`data` 534, 546-547
declaration 532, 536
enumeration 595
fill 542
floating-point 591-592
integral 592-594
NULL 600
numeric 547
`bitstring` 538, 547, 739
`byte` 538, 547, 739
`integer` 538, 547, 739
`longint` 538, 547, 739
point 542
ptrdiff_t 600
rectangle 542
redeclared 645
`resource` 534
`size_t` 600
string 540-541

`switch` 545
`unsigned char` 591
typecasting 93, 240

U

unary
inline functions 604
scope resolution operator 463
undoing changes to file 168

V

variable
changing values of 5, 85-86
`const` 727
examining 5, 80-86, 220-221
global 148, 210-211, 258, 285, 314, 657
definitions of 173
in code resource 277-278
in device driver 266-268
low-memory 359-361
in profile library 514-515
local 82, 508
aggregate initializer used for 268
macro 581. *See also* macros
QuickDraw 268, 279, 351
register 300-302, 384
resource compiler 563
setting 80-86
static 148
in code resource 277-278
in device driver 266, 268
string values of 563
temporary 722
THINK Rez 563
with dynamic storage 82, 237
virus checking 699-700
Visual Architect 5

W

warning messages 202-205
window
arranging 175-176, 671-672, 693
array 90, 241-242
Class Browser 289, 292
compile errors 59, 174, 201-205, 588
console 47-48, 97, 99, 401, 511, 588
data 72-73, 80-91, 220-221, 236-243, 683
clearing expression from 81, 86, 221
displaying contexts of expressions in 240
editing expressions in 238
entering expressions in 237-238
entering expressions more than once 86
parts of 81
removing expressions from 238
debugger 79-86, 101-102
drawing 97-99, 106-107
edit 41, 120, 125, 165, 174, 667-668
bringing to front 175
link errors 174, 201, 205, 403, 588
message 582-583
printing 174-175
project 39, 43, 46, 72, 100, 117, 123-126, 130, 148, 165, 174, 217, 668
bringing to front 81
colorizing 694
information shown in 148
search results 174
selecting 175-176
source 72-73, 76-79, 101, 219-228, 665
changing files in 79-80
context of expressions in 90
status panel in 73
struct 89, 90, 242-243
THINK Reference 405-406
Windows menu 671-672, 685
word wrap 168